

Analýza a syntéza řeči na mobilním zařízení pro zrakově postižené

Speech Analysis and Synthesis on Mobile Device for Visually Impaired

VŠB - Technická univerzita Ostrava
Fakulta elektrotechniky a informatiky
Katedra informatiky

Zadání diplomové práce

Student: **Bc. Antonín Brettšnajdr**
Studijní program: N2647 Informační a komunikační technologie
Studijní obor: 2612T025 Informatika a výpočetní technika
Téma: **Analýza a syntéza řeči na mobilním zařízení pro zrakově postižené**
Speech Analysis and Synthesis on Mobile Device for Visually Impaired

Zásady pro vypracování:

Cílem diplomové práce je návrh a realizace programu pro zpřístupnění základních funkcí mobilního telefonu lidem se zrakovým postižením.

1. Seznamte se s aplikacemi pro klienty se zrakovým postižením, problematikou analýzy a syntézy řeči, návrhem GUI pro zrakově postižené.
2. Analyzujte dostupné prostředky pro rozpoznání a syntézu řeči na platformě Android a navrhněte vhodné řešení.
3. Seznamte se s programovacím jazykem Java na mobilní platformě Android.
4. Implementujte aplikaci pro platformu Android a proveďte zhodnocení dosažených výsledků.
5. Porovnejte dosažené výsledky s existujícími aplikacemi, které se touto problematikou zabývají.
6. Vypracujte uživatelskou a programátorskou dokumentaci.

Seznam doporučené odborné literatury:

- [1] Reto Meier, Professional Android 2 Application Development, Wrox; 2 edition, 2010, ISBN-13: 978-0470565520
- [2] Bruce Eckel, Thinking in Java, Prentice Hall; 4 edition, 2006, ISBN-13: 978-0131872486
- [3] David Flanagan, Java In A Nutshell O'Reilly Media; 5th Edition, 2005, ISBN-13: 978-0596007737
- [4] Y. Daniel Liang, Introduction to Java Programming, Comprehensive, Prentice Hall; 8 edition, 2010, ISBN-13: 978-0132130806

Formální náležitosti a rozsah diplomové práce stanoví pokyny pro vypracování zveřejněné na webových stránkách fakulty.

Vedoucí diplomové práce: **doc. Ing. Lačezar Ličev, CSc.**

Datum zadání: 18.11.2011

Datum odevzdání: 04.05.2012



doc. Dr. Ing. Eduard Sojka
vedoucí katedry





prof. RNDr. Václav Snášel, CSc.
děkan fakulty

Místopřísežně prohlašuji, že jsem celou diplomovou práci vypracoval samostatně.

V Ostravě 3. května 2012

Bre/Mirinda.....

Souhlasím se zveřejněním této diplomové práce dle požadavků čl. 26, odst. 9 *Studijního a zkušebního řádu pro studium v magisterských programech VŠB-TU Ostrava*.

V Ostravě 3. května 2012

.....

Prohlašuji, že jsem tuto diplomovou práci vypracoval samostatně. Uvedl jsem všechny literární prameny a publikace, ze kterých jsem čerpal.

V Ostravě 3. května 2012

.....

Chtěl bych poděkovat panu doc. Ing. Lačezaru Ličevovi CSc. za rady ohledně této diplomové práce.

Abstrakt

Cílem této diplomové práce je vyvinout software pro usnadnění činností zrakově postižených lidí s pomocí technologií rozpoznávání řečového signálu a syntézy řeči na mobilním zařízení platformy Android. Jsou analyzovány matematické prostředky pro tvorbu a rozpoznávání lidského hlasového signálu včetně konkrétních dostupných nástrojů pro jejich realizaci pro operační systém Android. Dále jsou vybrány a analyzovány existující aplikace pro zrakově postižené, které jsou aktuálně dostupné na trhu. Ze získaných výsledků v analytické části je navrženo, implementováno a otestováno řešení na skutečném mobilním zařízení, které obsahuje funkce práce s SMS, kontakty, čtením textu a času, přehráváním mp3 a vyhledáváním MHD spojů ve městě Ostrava.

Klíčová slova: řeč, syntéza, rozpoznávání, Android, zrakově postižení

Abstract

The main goal of this thesis is to create a software solution, which will help to make the basic mobile device operations easily accessible by visually impaired people using the speech synthesis and recognition technology on Android platform. First of all an analysis of the mathematical approaches used for creating and recognizing a human speech signal is made, including available tools for Android platform. Next, actual existing applications for visually impaired on the market are chosen and analysed. From the results of the analytical part of this thesis, a solution is proposed, implemented and tested on a real mobile device, which includes fundamental functions as: SMS management, contact management, text reading, time, mp3 player and transport vehicles departure search inside the city of Ostrava.

Keywords: speech, synthesis, recognition, Android, visually impaired

Seznam použitých zkratk a symbolů

ACF	– Auto-Correlation Function
ADSL	– Asymmetric Digital Subscriber Line
API	– Application Programming Interface
CPU	– Central processing unit
CSS	– Cascading Style Sheets
DTW	– Dynamic Time Warping
GNU	– GNU's Not Unix
GPS	– Global Positioning System
GSM	– Groupe Spécial Mobile
GUI	– Graphical User Interface
HTML	– HyperText Markup Language
HTTP	– Hypertext Transfer Protocol
LPC	– Linear Predictive Coding
MHD	– Městská hromadná doprava
MP3	– MPEG-2 Audio Layer III
OCR	– Optical Character Recognition
OS	– Operating System
PC	– Personal Computer
PDA	– Personal Digital Assistant
POI	– Point Of Interest
SD	– Secure Digital
SDK	– Software Development Kit
SMS	– Short Message System
TTS	– Text To Speech
URL	– Uniform Resource Locator
UTF	– UCS Transformation Format
WCDMA	– Wideband Code Division Multiple Access

Obsah

1	Úvod	6
2	Lidská řeč	7
2.1	Model vytváření řeči	7
2.1.1	Proces vytváření lidské řeči	7
2.1.2	Samohlásky	7
2.1.3	Souhlásky	7
2.1.4	Modely pro analýzu a syntézu lidské řeči	10
2.2	Zpracování řečového signálu	12
2.2.1	Metody analýzy řečového signálu	12
2.2.2	Metody syntézy řečového signálu	15
3	Aplikace pro zrakově postižené	16
3.1	AT&T Mobile Accessibility Lite	16
3.2	TalkingPlaces	17
3.3	Voice Actions	19
4	Mobilní platforma Android	21
4.1	Hlasový vstup	21
4.2	SVOX TTS	22
4.3	Android TTS	23
5	Návrh řešení	25
5.1	Návrh řešení analýzy a syntézy řeči na Android zařízení	25
5.1.1	Volba softwarových a hardwarových prostředků	25
5.1.2	Syntéza řeči	26
5.1.3	Analýza řeči	28
5.2	Analýza a návrh základních funkcí programu	29
5.2.1	Hlasové příkazy a uživatelské rozhraní	29
5.2.2	Jízdní řady IDOS	29
5.2.3	Čas	31
5.2.4	Vytočení čísla	32
5.2.5	SMS	32
5.2.6	Kontakty	32
5.2.7	Čtení textů	33
5.2.8	Přehrávání mp3	33
6	Zhodnocení dosažených výsledků	35
7	Závěr	37
8	Reference	38

Přílohy	39
----------------	-----------

A Přílohy na CD-ROM	40
----------------------------	-----------

Seznam tabulek

1	Frekvence prvních dvou formantů českých samohlásek. [12]	9
2	Druhy úžinových souhlásek. [12]	9
3	Přehled českých souhlásek. [12]	9
4	Funkce všech 11 aplikací AT&T Mobile Accessibility Lite. [5]	16
5	Popis příkazů k akcím aplikace Voice Actions. [17]	20
6	Popis parametrů GET požadavku pro využití TTS modulu Google Překla- dače.	27
7	Hodnoty parametrů GET požadavku pro syntézu slova „Test“.	27
8	Ukázkový výpis druhů vyhledávání spojů a jejich URL adresy.	31
9	Popis argumentů pro získání výsledků hledání MHD spojů.	31
10	Srovnání připojení k internetu v závislosti na rychlosti odezvy syntézy řeči.	35

Seznam obrázků

2.1	Řečové orgány člověka. [12]	8
2.2	Grafické znázornění průběhu zvukového signálu slova „jiskřit“. [12] . . .	10
2.3	Ukázka posunu rámce řečového signálu při výpočtu autokorelační funkce. [6]	13
2.4	Výsledná autokorelační funkce řečového segmentu. [6]	13
2.5	Určený lag autokorelační funkce pomocí nalezení maxima (v tomto případě $L=87$). [6]	14
2.6	Model produkce lidské řeči. [11]	15
3.7	Hlavní menu aplikace AT&T Mobile Accessibility Lite. [4]	17
3.8	Ukázka obrazovky hlavního menu a zobrazení mapy v aplikaci Talking Places. [16]	18
4.9	Integrace hlasového vstupu v aplikaci Handcent SMS. [2]	22
5.10	Schéma funkce převodu textu na řeč pomocí TTS služby firmy Google. . .	28
5.11	Schéma rozvržení a propojení jednotlivých stránek aplikace.	30

Seznam výpisů zdrojového kódu

1	Kód pro integraci funkce hlasového vstupu do Android aplikace. [2] . . .	22
2	Kód spuštění intentu pro ověření dostupnosti požadovaného jazyka pro TTS engine. [3]	23
3	Kód pro stažení a instalaci chybějícího jazyka. [3]	23
4	Ukázka použití metody <i>speak()</i> . [3]	24

1 Úvod

Nacházíme se v době, kdy technický pokrok jde velmi rychlým tempem kupředu. Významným úspěchem nebyly pouze stolní počítače, ale taky silný technologický růst v oblasti mobilních zařízení. Tyto zařízení rozšiřují pole působnosti technologií jako jsou syntéza řeči a rozpoznávání řeči, čímž otevírají nový způsob jejich využití u lidí se zhoršenou schopností vidět. Měli-li bychom říct, který ze způsobů komunikace je pro nás nejdůležitější a zároveň nejpřirozenější, určitě se shodneme na tom, že je to právě řeč. Už po tisíciletí probíhá převážná část komunikace mezi lidmi formou mluveného slova, avšak dnes se tato potřeba rozšiřuje i o oblast komunikace mezi člověkem a strojem, a to především s počítačem. [12]

Stále více oblíbeným mobilním zařízením se začíná stávat PDA, které disponuje vlastním operačním systémem a je v zájmu velkých firem vyvinout podstatně sofistikovanější softwarové prostředky než u jejich předchůdců - mobilních telefonů. Ovládání počítače nebo zařízení PDA hlasem ulehčí mnoha lidem celkový proces učení, jak dané zařízení ovládat. Nejen že mnoho lidí (především staršího věku) přistupuje k těmto produktům moderní doby s určitým odstupem a nedůvěrou, ale pro lidi se zhoršeným zrakem je to jediná možnost, jak se s těmito přístroji dorozumět.

Cílem této práce je návrh a realizace programu pro ulehčení ovládání základních funkcí mobilního zařízení PDA s operačním systémem Android pro lidi se zrakovým postižením a to s pomocí technik analýzy a syntézy řeči. Je navrženo a realizováno řešení, které pomůže lidem se zhoršeným zrakem provádět na svém mobilním zařízení za pomoci jednoduchého uživatelského rozhraní, ovládané dotykem a hlasem uživatele, tyto akce: odesílat a číst SMS, číst uložené texty ve formátu TXT, přehrávat MP3 soubory, vytvářet a mazat kontakty, vytočit kontakt nebo telefonní číslo, hledat spoje MHD ve městě Ostrava, přechíst aktuální datum a čas.

První část této práce se zabývá samotnou problematikou lidské řeči obecně s detailním pohledem na techniky analýzy a syntézy řeči, které se k těmto účelům používají. V další části jsou popsány některé již existující profesionální aplikace určené pro uživatele se zrakovým postižením. Ve čtvrté kapitole jsou probrány dostupné prostředky pro syntézu a rozpoznávání řeči na platformě Android. V páté kapitole předkládám analýzu a navrhované řešení výsledné aplikace. V následujících dvou kapitolách jsou shrnuty výsledky navržené aplikace v porovnání s existujícími aplikacemi zabývajícími se podobnou problematikou a závěr.

2 Lidská řeč

2.1 Model vytváření řeči

2.1.1 Proces vytváření lidské řeči

Zvuk mluveného slova vzniká v lidských řečových orgánech, které se skládají z hlasivek, dutiny hrdelní, ústní a nosní, měkkého a tvrdého patra, zubů a jazyka. Tyto řečové orgány jsou schématicky znázorněny na obr. č. 2.1. Dále je třeba vzít v úvahu taky plíce, které jsou zdroj energie pro vznik zvukových kmitů a bez nich by nebyla komunikace mluveným slovem možná. [12]

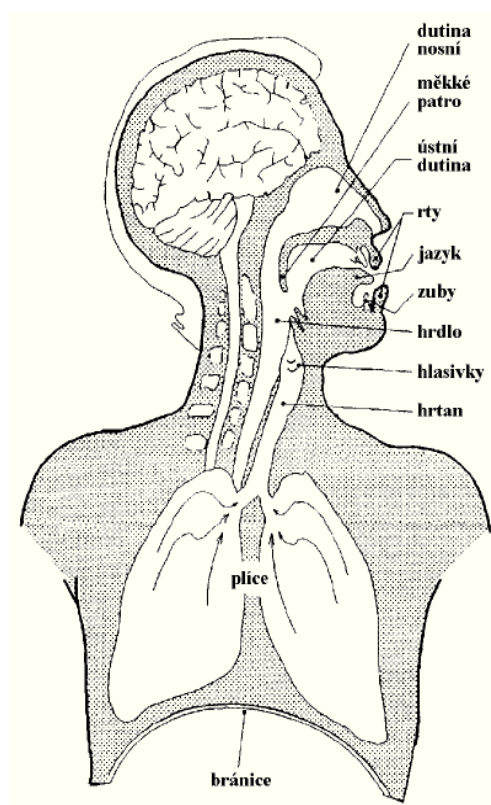
Všechny znělé zvuky vznikají za pomoci hlasivek, které se nachází v horní části hrtanu. Prostor mezi hlasivkami pak tvoří tzv. hlasivkovou štěrbinu. Pokud člověk nemluví, je hlasivková štěrbinu za pomoci chrupavek odkrytá a nebrání tak ve volném průchodu vzduchu. Pokud člověk začne mluvit, jeho hlasivky se začnou svírat a roztažovat, čímž dají vznik jednotlivým řečovým zvukům. Za působení tlaku vzduchu z plic se stažené hlasivky stanou pružnými a začnou kmitat. Frekvence těchto kmitů závisí na tlaku vzduchu a svalovém napětí hlasivek. Děti a dospělí, muži a ženy, mají různou frekvenci základního tónu jejich hlasivek, avšak u většiny lidí je tato frekvence v rozmezí 150Hz - 400Hz. Tato frekvence se označuje symbolem F_0 a představuje tzv. základní tón lidského hlasu. Základní tón je přítomen při vzniku všech znělých zvuků, tj. samohlásek a znělých souhlásek. Vzniklý zvuk se dále šíří hlasivkovým traktem a je vyzařován rty do volného prostoru. [12]

2.1.2 Samohlásky

Když člověk vysloví samohlásku, vzduch proudí z jeho plic přes hlasové ústrojí velmi volně. V akustickém spektru výsledného zvuku je obsažen základní tón hlasu a jeho další vyšší frekvence, které vznikají rezonancí v dutině hlasového traktu. Fonetika nazývá tyto vyšší rezonující frekvence formanty a označuje je čísly od 0 až po n , tj. $F_1, F_2, F_3, \dots, F_n$, kde F_0 odpovídá základní frekvenci hlasivek člověka. V Českém jazyce jsou pro samohlásky nejdůležitější formanty F_1 a F_2 , jejichž výška a intenzita závisí na uspořádání, délce, tvaru a průřezu dutiny ústní i hrdelní. Výška a intenzita těchto rezonančních frekvencí je dále měněna průchodem přes rty, čelisti a měkké patro, které uzavírá vstup proudu vzduchu do nosní dutiny (platí pro všechny české samohlásky). České samohlásky a jejich formantové frekvence jsou uvedeny v tabulce č. 1. [12]

2.1.3 Souhlásky

Souhlásky se od samohlásek odlišují především přítomností charakteristického šumu v jejich akustickém spektru hlásek. Souhlásky vznikají vzduchovými turbulencemi, které se vytvářejí třením vzduchu o artikulační orgány během výdechu tvořící překážku proudu vzduchu (např. jazyk, zuby, rty). Překážky se dělí na dva druhy, a sice úplné (okluzívy) nebo částečné (frikativy). Úplná překážka způsobuje v prvotní fázi hroma-



Obrázek 2.1: Řečové orgány člověka. [12]

dění tlaku vzduchu jdoucím z plic a následně po jejím uvolnění vytváří charakteristický šum, který se podobá explozi. Mezi tento druh souhlásek patří české [12]

$p, t, t', k, b, d, d', g, m, n, ň$

Pokud je překážka částečná, zvuk takové souhlásky vzniká třením o zúženou část artikulačního ústrojí, čímž vzniká specifický třecí šum. Tyto souhlásky dělíme na vlastní úžinové, bokové a kmitavé, kde konkrétní příklady těchto samohlásek jsou uvedeny v tabulce č. 2. Existuje malá skupina souhlásek, které jsou tvořeny kombinací obou typů překážek v hlasovém ústrojí. Tyto souhlásky nazýváme **polozávěrové** a patří mezi ně české *c* a *č*. [12]

Dalším kritériem pro třídění souhlásek je jejich znělost. Pokud člověk vysloví neznělou souhlásku, jeho hlasivky jsou oddáleny, tudíž nevytvářejí základní hlasový tón, ale pouze propouští vzduch tak, jak je tomu u běžného dýchání. Naopak v případě znělých souhlásek je při jejich vytváření přítomen základní tón hlasivek. V případě většiny souhlásek uzavírá měkké patro cestu z dutiny hrdelní do dutiny nosní, která se zúčastňuje artikulace až v momentě uvolnění měkkého patra, např. při vyslovení českých *m*, *n* a *ň*. Většina šumových souhlásek může být seřazena do dvojic, které se shodují svým způsobem artikulace, ale liší se v jejich znělosti. Takové souhlásky se nazývají párové. Naopak

Samohláska	F ₁	F ₂
u	300 - 500 Hz	600 - 1000 Hz
o	500 - 700 Hz	900 - 1200 Hz
a	750 - 1100 Hz	1100 - 1500 Hz
e	500 - 700 Hz	1500 - 2000 Hz
i	300 - 500 Hz	2000 - 3000 Hz

Tabulka 1: Frekvence prvních dvou formantů českých samohlásek. [12]

Typ	Příklad
vlastní úžinové	<i>f, v, s, š, z, ž, j, ch, h</i>
bokové	<i>l</i>
kmitavé	<i>r, ř</i>

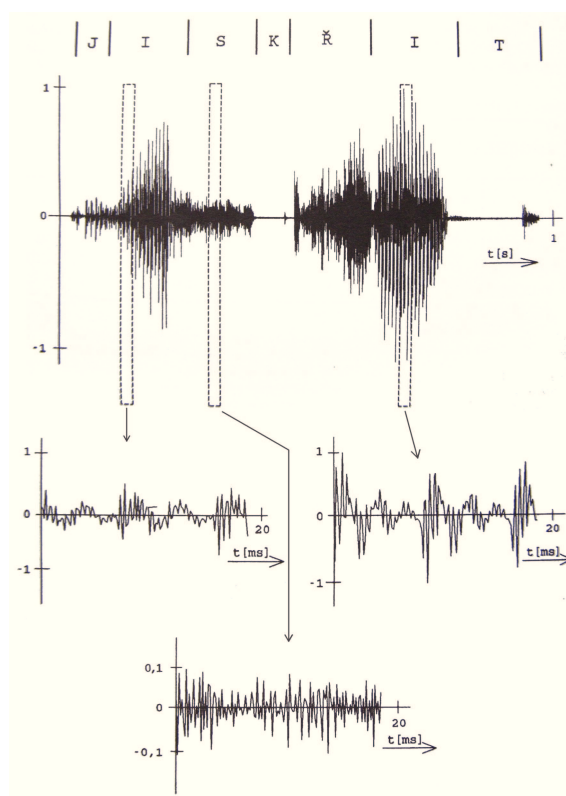
Tabulka 2: Druhy úžinových souhlásek. [12]

souhlásky nepárové nemají svůj neznělý protějšek a jsou vždy znělé. Kompletní přehled českých souhlásek je uveden v tabulce č. 3. [12]

Souhlásky		Závěrové	Úžinové	Polozávěrové
Párové	neznělé	<i>p, t, t', k</i>	<i>s, š, f, ch</i>	<i>c, č</i>
	znělé	<i>b, d, d', g</i>	<i>z, ž, v, h</i>	<i>dz, dž</i>
Nepárové	znělé	<i>m, n, ň</i>	<i>l, j, r, ř</i>	

Tabulka 3: Přehled českých souhlásek. [12]

Aby mohl být zvuk hlasovým ústrojím vysloven, musí nejdříve určité části hlasového ústrojí zaujmout počáteční polohu a během doby vyslovování zvuku svou polohu měnit podle předem určeného programu. Není však možné aby artikulační orgány měnily svou polohu okamžitě, ale tato operace zabere jistý čas, který je závislý na hmotnosti jednotlivých artikulačních orgánů a taky na síle svalů, které těmito orgány pohybují. Postup zpracování řečového signálu komplikuje skutečnost, že se způsob vyslovení jednoho fonému může měnit v závislosti na jeho kontextu ve slově, tzn. různí se jeho akustický doprovod. Tento jev se nazývá **koartikulace** a z hlediska fonému je závislý na zvuku předešlého i následujícího fonému, ale taky na tempu a intonaci řeči. Pro představu komplexity problému vezměme v úvahu určitý foném. Pak zvukových variací je právě tolik, kolik je přípustných sousedních fonémů. Ukázka odlišnosti zvuku stejného fonému v závislosti na kontextu je znázorněna na obr. č. 2.2. [12]



Obrázek 2.2: Grafické znázornění průběhu zvukového signálu slova „jiskřit“. [12]

2.1.4 Modely pro analýzu a syntézu lidské řeči

Zvuk lidské řeči vzniká troji způsobem, a sice znělými zvuky, frikativními zvuky a explozivními zvuky. **Znělé zvuky** vznikají za přítomnosti základního tónu hlasivek, které se rozvibrují za pomoci energie stlačeného vzduchu. **Frikativní zvuky** jsou tvořeny turbulencemi proudícího vzduchu, který je protlačován mezi štěrbinami artikulačních orgánů a vytváří tak charakteristický šum. **Explozivní zvuky** vznikají uzavřením proudu vzduchu a jeho nahromaděním. Samotný zvuk vznikne až po náhlém uvolnění, která vytvoří krátkou explozi v akustickém spektru šumu. Většina zvuků lidského hlasu vzniká kombinací výše uvedených způsobů. [12]

Hlavním cílem modelování produkce řeči je nalezení reprezentace akustických fyzikálních jevů pomocí matematických vztahů s požadavkem na jednoduchost a maximální možnou přesnost matematického modelu. Jelikož během mluvení nejsou hlasivky zcela odpojeny od hlasového traktu, obsahuje lidská řeč nelineární charakteristiky, což přidává matematickým modelům lidské řeči na složitosti. Z tohoto důvodu je možné sestavit lineární a časově nezávislý matematický model lidské řeči, který však bude platný pouze v krátkých časových intervalech. U mnohých zvuků lidské řeči můžeme počítat s tím, že typ buzení a vlastnosti hlasového traktu budou téměř neměnné pro časový úsek v

rozmezí 10ms až 20ms. Vzhledem k této skutečnosti by se výsledný model lidské řeči mohl skládat z lineárního modelu hlasového traktu s pomalu se měnícími parametry, buzený vhodným budícím signálem. V případě znělého signálu bude model buzen sledem jednotkových impulsů o periodě T_0 a v případě neznělého signálu bílým šumem.[12]

Model hlasivek

Pro vytvoření modelu hlasivek budou klíčové následující parametry: tlak vzduchu z plic, hmotnost hlasivek, pružnost hlasivek a velikost plochy hlasivkové štěrbin v uvolněném stavu. Na těchto parametrech závisí výsledná frekvence hlasivek, která bude produkována v případě znělých zvuků. Pro modelování rychlosti kmitání hlasivek použijeme nízkopásmový filtr druhého řádu, jehož parametry budou pro každého řečníka specifické. Přenosová funkce modelu hlasivek je vyjádřena ve vzorci (2.1), kde c je neznámý parametr filtru a T je perioda vzorkování. [12]

$$G(z) = \frac{1}{[1 - \exp(-cT)z^{-1}]^2} \quad (2.1)$$

Model hlasového traktu

V případě modelování hlasového traktu se zaměříme na dutinu hrdelní, dutinu ústní a taky dutinu nosní. Parametry tohoto modelu budou záviset na průřezu a délce těchto dutin, kde ve většině případech se budou lišit jak v případě různých lidí, tak i v rámci stejného člověka při vyslovování různých zvuků. Výsledný zvuk může taky ovlivnit pozice jazyka atd. Výsledný model hlasového traktu bude tvořen malým počtem dvoupólových rezonátorů uspořádaných kaskádovitě. Frekvence těchto rezonátorů budou odpovídat frekvencím formantů. Přenosová funkce celopólového filtru je uvedena ve vzorci (2.2): [12]

$$V(z) = \frac{1}{\prod_{i=1}^K [1 - 2\exp(-\alpha_i T)\cos(\beta_i T)z^{-1} + \exp(-2\alpha_i T)z^{-2}]} \quad (2.2)$$

kde K je počet formantů modelu (pro většinu zvuků Českého jazyka postačí 3 až 4 formanty), α_i , resp. β_i , jsou neznámé koeficienty modelu a T je perioda vzorkování. [12]

Model vyzářování zvuku

Model fyzikálních jevů, které ovlivňují podobu výsledného zvuku, je odvozen vztahem: [12]

$$L(z) = 1 - z^{-1} \quad (2.3)$$

Abychom dostali celkovou podobu modelu vytváření lidské řeči, sjednotíme předchozí modely do jednoho vztahu, který se po zkrácení zjednoduší do tvaru přenosové funkce $H(z)$: [12]

$$H(z) = \frac{1}{1 + \sum_{i=1}^Q a_i z^{-i}} \quad (2.4)$$

kde $Q = 2K + 1$ a K je počet formantů, které chceme modelem zachytit (3 až 4). Pokud chceme, aby model kvalitněji zachycoval nosovky a frikativní zvuky, bude potřeba zvolit Q zpravidla vyšší. [12]

2.2 Zpracování řečového signálu

2.2.1 Metody analýzy řečového signálu

Zpracování řeči je multi-disciplinární obor, který zahrnuje oblasti technické, humanitní i přírodní. Předmětem zájmu tohoto oboru je jak samotná komunikace člověka s člověkem, tak i komunikace člověka se strojem. [6]

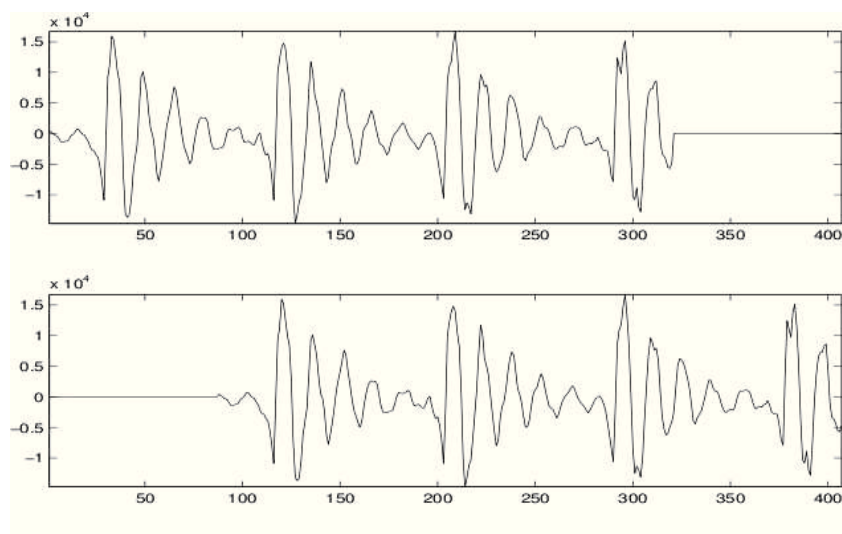
Pro vytvoření konkrétních metod analýzy řečového signálu je nejdříve nutné porozumět **fyzilogii** hlasového a sluchového ústrojí, na základě které se sestaví matematické modely. Tvorbou a slyšením hlasového signálu se zabývá **akustika**, která je důležitá v oblastech analýzy i syntézy řeči. Na základě modelů hlasového a sluchového ústrojí s aplikací poznatku akustiky jsou navrženy metody **zpracování signálu**, které sestávají z modelování, spektrální analýzy, teorie informace, rozpoznávání vzorů, kódování aj. Z oblasti **humanitních věd** se pak podrobněji v oblasti zpracování řečového signálu zabývá **fonetika** (tvorba a slyšení zvuku řečového signálu), **fonologie** (fonémy a jejich systém v cílovém jazyce), **prosodie** (intonace slov a vět), **lexikologie** (slovní zásoba jazyka a jejich vzájemný vztah a vývoj), **gramatika** (pravidla obměn mezi slovy a jejich spojování uvnitř vět), **syntaxe** (gramatika vět a souvětí) a **sémantika** (výklad významu vět). [6]

Autokorelační funkce (ACF)

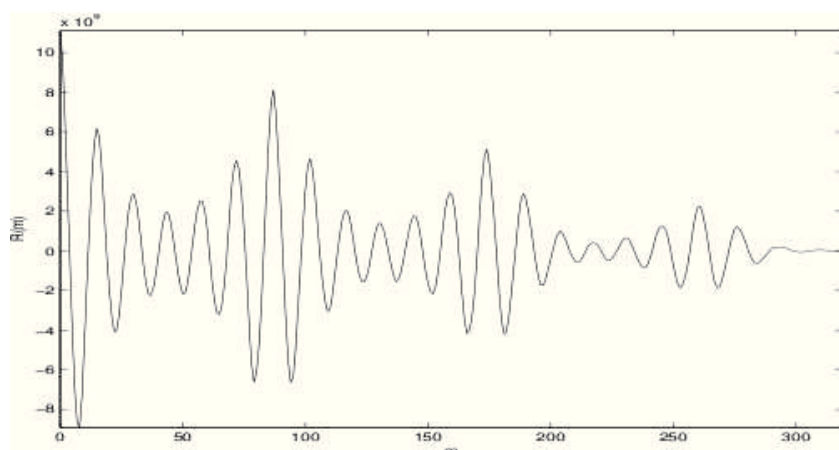
Krátkodobá autokorelační funkce porovnává signál se sebou samým a používá se k učení periodičnosti signálu. Z výsledného grafu je pak taky možné určit základní tón hlasivek analyzovaného segmentu řečového signálu. Vztah pro výpočet krátkodobé autokorelační funkce je definován [6]:

$$R(m) = \sum_{n=0}^{N-1-m} s(n)s(n+m) \quad (2.5)$$

Ilustrace posunu řečového rámce je znázorněna na obr. 2.3. Po aplikaci všech posunů dle výše uvedeného vzorce dostaneme hodnoty výsledné autokorelační funkce pro daný řečový segment, která je vykreslena na obr. 2.4. Ve výsledném grafu autokorelační funkce se pak hledá tzv. **lag**, pomocí kterého určíme základní tón analyzovaného segmentu řečového signálu. Lag je definován jako maximum v určité oblasti grafu autokorelační funkce a je v hodné ho hledat ve frekvenčním rozpětí 100Hz (mužský hlas) až 500Hz (ženský nebo dětský hlas). Lag vypočteme podle vzorce (2.6). [6]



Obrázek 2.3: Ukázka posunu rámce řečového signálu při výpočtu autokorelační funkce. [6]



Obrázek 2.4: Výsledná autokorelační funkce řečového segmentu. [6]

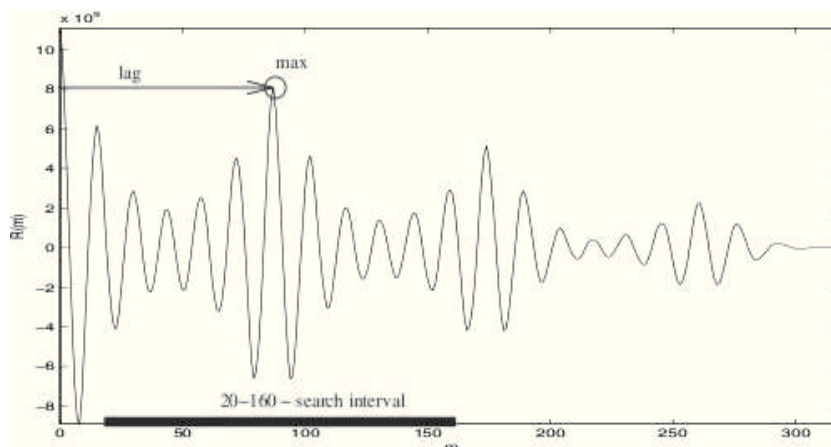
$$lag = arg \max_r R(m) \quad (2.6)$$

Hodnotu lagu lze taky použít pro určení znělosti analyzovaného segmentu řeči, tj. jestli je v daném segmentu přítomen základní tón hlasivek (znělý segment), anebo známky základního tónu nejsou patrné (neznělý segment). Znělost segmentu analyzovaného řečového signálu pak určíme porovnáním hodnoty nalezeného maxima s maximální hodnotou autokorelační funkce, které odpovídá nultému koeficientu (též se nazývá energií analyzovaného segmentu řeči). Ve vzorci bude figurovat konstantu α , která slouží ke kalibraci vzorce vůči specifickým podmínkám analýzy řečového signálu a je

nutné ji zvolit experimentálně. Vzorec pro určení znělosti analyzovaného segmentu řeči je definován [6]:

$$\begin{aligned} R_{max} < \alpha R(0) &\Rightarrow \text{unvoiced} \\ R_{max} &\geq \alpha R(0) \Rightarrow \text{voiced} \end{aligned} \quad (2.7)$$

Pro uvedený příklad jsou nalezený lag a velikosti koeficientů R_{max} a $R(0)$ zobrazeny na obr. 2.5. [6]



Obrázek 2.5: Určení lag autokorelační funkce pomocí nalezení maxima (v tomto případě $L=87$). [6]

Pro praktické účely je potřebné přepočítat hodnotu lagu do časové nebo frekvenční oblasti, čímž dostaneme konkrétní hodnotu základního tónu v analyzovaném segmentu řečového signálu a to buď v sekundách, anebo v Hz. Přepočet provedeme podle následujícího vzorce, kde F_s je vzorkovací frekvence, T_0 je perioda základního tónu a L je hodnota lagu [6]:

$$L = T_0 F_s \quad (2.8)$$

Krátkodobá intenzita

V oblasti analýzy řečového signálu je v prvotní fázi nahraného záznamu odlišit tiché části od částí promluvy. Pro tyto účely se používá krátkodobá intenzita, která se podobně jako autokorelační funkce počítá pro jednotlivé segmenty řečového signálu. Hodnota krátkodobé intenzity se vypočítá podle následujícího vzorce [12]:

$$M_n = \sum_{k=-\infty}^{\infty} |s(k)| w(n-k) \quad (2.9)$$

Další uplatnění krátkodobé intenzity je v rozpoznávání znělosti analyzovaných segmentů řeči. [12]

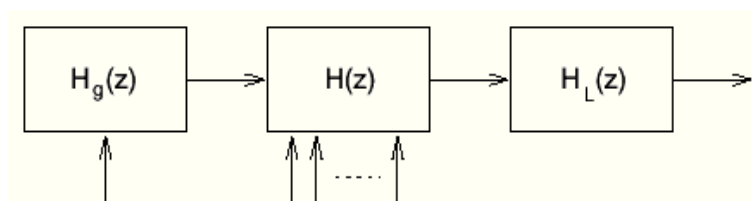
Dynamické borcení času (DTW)

Tato metoda se používá pro srovnávání dvou segmentů řeči v časové oblasti a následné vyhodnocení jejich podobnosti. Zpracovávané segmenty se mohou lišit čase nebo rychlosti, a proto se tato technika využívá v oblasti identifikace osob podle stylu chůze zaznamenané např. na kamerovém zařízení. Pro aplikaci metody dynamického borcení času je nutné mít vytvořenou referenční banku vzorů, se kterými budou nahrané segmenty řeči porovnávány. To znamená, že na vstupu máme matici parametrů analyzovaného segmentu řeči v časové oblasti a chceme zjistit, ke které matici parametrů naší banky tento vzor patří. [6]

2.2.2 Metody syntézy řečového signálu

Formantová syntéza

Metoda formantové syntézy patří mezi zástupce, které operují ve frekvenční oblasti. Jsou založené na generování základního tónu hlasu a dále modelování jeho rezonančních frekvencí (formantů). Tento druh syntézy řeči se mezi všemi jinými nejvíce podobá funkci skutečného hlasového traktu. Formantový syntetizér nemodeluje hlasové ústrojí do všech detailů, ale pro vytvoření výsledného zvukového signálu používá několik jednoduchých pravidel, které vychází z teorie vytváření lidské řeči. Zdroj hlasového signálu je ve frekvenční oblasti popsán svým spektrem $H_g(z)$, a jeho funkcí je buzení model hlasového traktu, který je definován přenosovou funkcí $H(z)$. Přenosová funkce se skládá z několika rezonátorů, kde výslednému zvukovému signálu dá finální podobu model vyzařování řeči do prostoru $H_L(z)$. Model produkce lidské řeči je znázorněn na obr. č. 2.6. [11]



Obrázek 2.6: Model produkce lidské řeči. [11]

Syntéza na bázi lineární predikce

Dalším ze zástupců syntézy řeči ve frekvenční oblasti je metoda syntézy založená na bázi lineárního prediktivního kódování (LPC). Tento způsob kódování je založený na odhadu parametrů filtru, který modeluje artikulační trakt. V analytické části se při vytváření fonetických jednotek nejprve zjišťují parametry filtru, znělost řečového signálu, a v případě, že je řečový signál znělý, tak i jeho perioda a intenzita. Syntéza výsledného signálu se provádí buzením LPC filtru impulsovým signálem o dané frekvenci a intenzitě v případě znělého signálu a bílým šumem v případě neznělého signálu. [6]

3 Aplikace pro zrakově postižené

3.1 AT&T Mobile Accessibility Lite

AT&T Mobile Accessibility Lite je mobilní aplikace přístupnosti od firmy Code Factory, která je navržena pro pomoc lidem se zrakovým postižením, aby mohli využívat více služeb jejich Android zařízení. Tato aplikace je dostupná zdarma na omezenou dobu pro zrakově postižené lidi, kteří používají Android verze 2.1 nebo novější. [5]

AT&T Mobile Accessibility Lite se skládá z 11 aplikací přístupnosti, které byly speciálně navrženy pro lidi trpící slepotou nebo zrakovým postižením. Všechny mají zjednodušené uživatelské rozhraní, jehož text je automaticky syntetizován pomocí Nuance Vocalizer. Tyto aplikace umožňují zrakově postiženým vykonávat: [5]

Aplikace	Funkce
Phone	Volání, příjem hovorů, ID volajícího a správa historie hovorů.
Contacts	Správa kontaktů včetně kontaktů na sociálních sítích jako např. Facebook.
SMS	Psaní a čtení SMS, správa konverzací.
Alarms	Nastavení budíku.
Web	Kompletní zkušenost procházení webu podobně jako na PC, rychlá navigace podle prvků dle volby (odkazy, odstavce, nadpisy, formuláře atd.), bookmarky.
Calendar	Vytváření, úprava a mazání kalendářových záznamů. Prohlížení všech událostí za den, týden nebo měsíc.
Email	Plný přístup ke Gmail účtu.
Where am I?	GPS aplikace, která zjišťuje aktuální pozici.
Accessible Music Player	Přehrávač s intuitivním rozhraním, který automaticky načte všechnu hudbu z paměťové karty telefonu a seřadí ji podle umělce a alba.
Apps	Přístup k seznamu všech nainstalovaných aplikací na telefonu.
Settings	Změna vyzváněcího tónu, nastavení upozornění (vibrace nebo zvuk), nastavení ozvěny klávesnice, obsáhlost interpunkce, rychlost syntézy řeči, atd.

Tabulka 4: Funkce všech 11 aplikací AT&T Mobile Accessibility Lite. [5]

AT&T Mobile Accessibility Lite dále nabízí tyto funkce [5]:

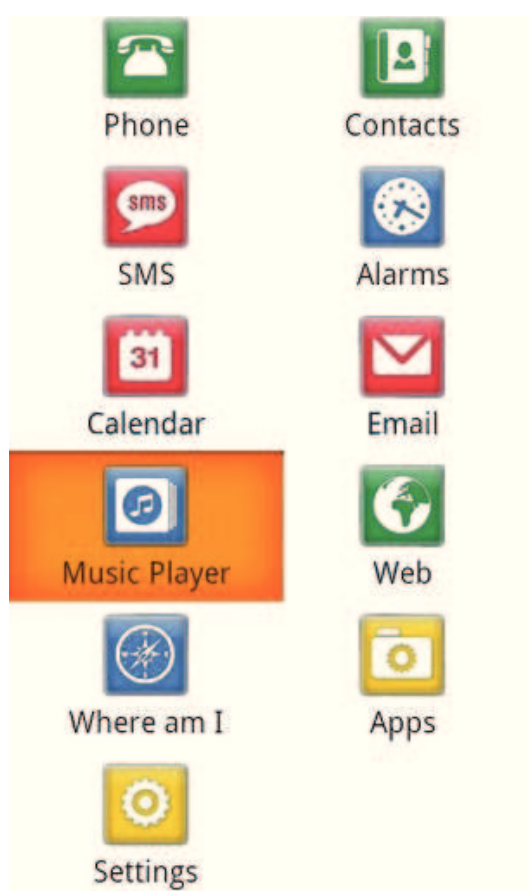
Dotyková navigace: Tato vstupní metoda umožňuje použít obrazovku PDA jako klávesnici. Uživatel pohybuje prstem po obrazovce a syntéza řeči vždy přečte, na kterém

písmenu se nachází. Po nalezení správného písmene uživatel uvolní svůj prst od obrazovky a písmeno se přidá k textu SMS. Pro uživatele, kteří preferují výraznější odezvu telefonu, je možné zapnout vibrace.

Jednoduché vkládání textu: Vkládat text lze kromě qwerty klávesnice taky přímo pomocí rozpoznávání řeči. Uživatel slovy nadiktuje obsah SMS, který se přímo převede na text.

Syntéza řeči: AT&T Mobile Accessibility Lite používá 2 syntetizéry řeči, a sice Nuance Vocalizer a Android Text-To-Speech API. Druhá varianta umožňuje uživateli používat již nainstalované hlasy na jeho telefonu, včetně předinstalovaných.

Celá sada aplikací podporuje celkem 9 jazyků, mezi které patří: **angličtina, španělština, němčina, italština, francouzština, portugalština, čeština, holandština a ruština.**



Obrázek 3.7: Hlavní menu aplikace AT&T Mobile Accessibility Lite. [4]

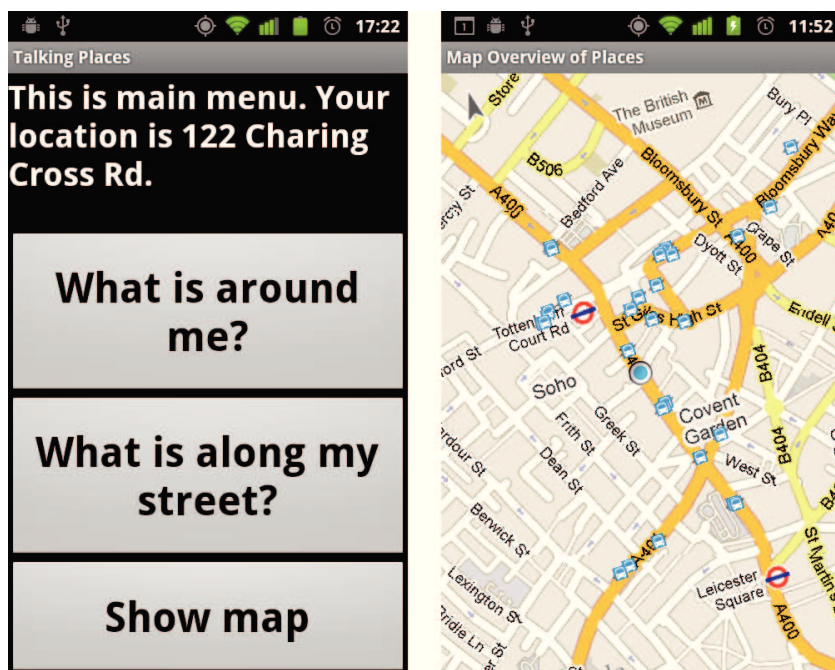
3.2 TalkingPlaces

Aplikace Talking Places byla navržena v rámci diplomové práce na FEL ČVUT v Praze. Je zaměřena na slepé lidi, ale je taky vhodná pro lidi bez zrakového postižení. Hlavní

funkcí této aplikace je pomocí syntézy řeči informovat o tzv. „bodech zájmu”(POI) v blízkém okolí, kde se aktuálně uživatel nachází. Těmito body zájmu jsou například zastávky MHD, restaurace, místa z Wikipedie atd. Jako zdroj dat je použit server OpenStreetMap. Mezi základní funkce této aplikace patří [16]:

- přístupnost nevidomým uživatelům díky dobré spolupráci s aplikacemi SoundBack, TalkBack a KickBack
- detekce bodu zájmu (POI) v okolí pomocí GPS
- zdroj dat z Panoramio, Google Maps, Wikipedia a OpenStreetMap
- seřazení a filtrování výsledků
- filtrování POI podle aktuální ulice, kde se uživatel nachází
- hmatová mapa míst
- vyhledávání míst podle jména
- navigační pokyny pro chůzi pěšky z Google Maps

Textové prvky uživatelského rozhraní, dat i syntézy řeči jsou v **anglickém** jazyce. Ukázka uživatelského rozhraní je znázorněna na obrázku 3.8.



Obrázek 3.8: Ukázka obrazovky hlavního menu a zobrazení mapy v aplikaci Talking Places. [16]

3.3 Voice Actions

Další aplikací pro ulehčení ovládání základních úkonů na Android zařízení je Voice Actions od společnosti Google. Voice Actions je součástí aplikace pro vyhledávání na vyhledávací firmy Google pomocí rozpoznání hlasu. Aplikace je navržena tak, aby uživatel mohl spustit požadovanou akci pomocí jednorázového vstupního hlasového příkazu. V případě posílání SMS, emailu a zápisu poznámek je dále možné přidávat další věty (opět pomocí rozpoznání hlasu). Voice Actions podporuje tyto akce: SMS, hudební přehrávač, navigace, email, mapa, webová stránka, poznámky a vyhledávání Google. Popis jednotlivých akcí včetně příkazů pro jejich spuštění je uveden v tabulce 5. [17]

Příkazy k ovládání akcí této aplikace jsou dostupné v **anglickém jazyce**. I přesto že aplikace Voice Actions nebyla navržena přímo pro uživatele se zrakovým postižením, některé její funkce (jako např. přehrávání hudby, poznámky, vytáčení kontaktů aj.) by pro zrakově postižené uživatele mohly znamenat výrazné ulehčení. [17]

Akce	Příkaz
Zasílání textových zpráv	<i>"send text to [recipient] [message]*"</i> Např.: <i>"send text to Allison Miller Running late. I will be home around 9"</i>
Poslech hudby	<i>"listen to [artist/song/album]"</i> Např.: <i>"listen to the decemberists"</i>
Navigace	<i>"navigate to [address/city/business name]"</i> Např.: <i>"navigate to the DeYoung Museum San Francisco" or "navigate to 1965 Page St."</i>
Vytočit firmu	<i>"call [business name] [location]*"</i> Např.: <i>"call Pizzeria Venti Mountain View"</i>
Vytočit kontakt	<i>"call [contact name] [phone type]*"</i> Např.: <i>"call Allison Miller home"</i>
Odeslat email	<i>"send email to [recipient] [subject]* [body]*"</i> Např.: <i>"send email to Mike LeBeau How's life in New York treating you? The weather's beautiful here!"</i>
Zobrazit mapu	<i>"map of [address/city]"</i> Např.: <i>"map of San Francisco"</i>
Otevřít web	<i>"go to [website]"</i> Např.: <i>"go to Wikipedia"</i>
Vytvořit poznámku	<i>"note to self [message]"</i> Např.: <i>"note to self grocery list banana milk eggs pizza"</i>
Vyhledávání Google	<i>"[your query]"</i> Např.: <i>"pictures of the golden gate bridge at sunset"</i>

Tabulka 5: Popis příkazů k akcím aplikace Voice Actions. [17]

4 Mobilní platforma Android

Android je rozsáhlá open source platforma, která byla navržena především pro mobilní zařízení (SmartPhones, PDA, tablety, navigace). Celá platforma běží na operačním systému založeným na jádru Linuxu. Platforma Android dává k dispozici nejen operační systém s uživatelským prostředím pro koncové uživatele, ale taky kompletní řešení nasazení operačního systému (specifikace driverů aj.) pro mobilní operátory a výrobce zařízení. Dále poskytuje vývojářům aplikací efektivní nástroje pro jejich vývoj - tzv. Software Development Kit (SDK) [1].

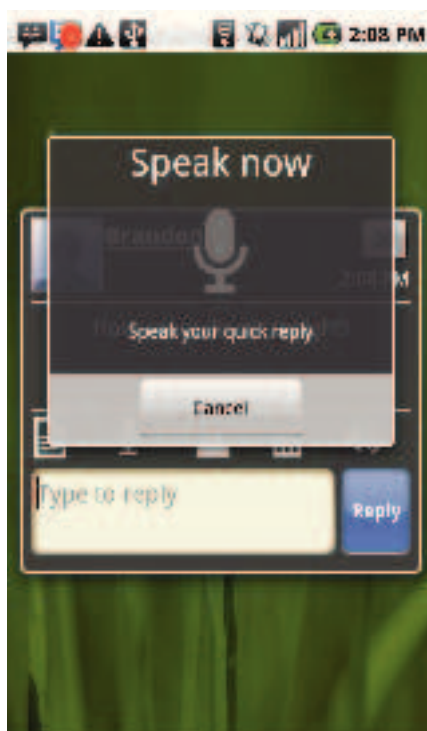
Nástroje pro vývoj aplikací na platformě Android jsou obsaženy v SDK, který je dostupný pro všechny hlavní platformy operačních systémů GNU/Linux, Windows i Mac OS. Sada SDK je rozdělena na tři druhy: základní (nástroje nezbytné pro vývoj aplikací), doporučená a plná konfigurace vývojového prostředí. Pro simulaci svých aplikací může vývojář použít emulátor, který je součástí Android SDK. Emulátor umožňuje vyzkoušet Android aplikaci bez nutnosti vlastnit fyzické zařízení. [1].

4.1 Hlasový vstup

Většina Android zařízení obsahuje předinstalovanou aplikaci Voice Search firmy Google, která se používá hlasové vyhledávání pomocí vyhledávače Google a dále taky jako Voice Actions, které jsou popsány blíže v kapitole 3.3. Rozpoznávání hlasu dosahuje vysoké kvality, je dostupné v mnoha světových jazycích a pro využívání této služby je potřeba mít dostupné připojení k internetu. V průběhu vývoje byl rozsah využití hlasového vstupu rozšířen a od verze Androidu 2.1 byla představena „hlasová klávesnice“, která umožnila uživatelům mobilních zařízení s operačním systémem Android diktovat zprávy hlasem, namísto aby je museli psát ručně pomocí klasické virtuální klávesnice na dotykové obrazovce. [2]

Android SDK umožňuje vývojářům integrovat funkce hlasového vstupu přímo do jejich aplikace. Jedna z aplikací na Google Play, která integraci funkce hlasového vstupu využívá, je Handcent SMS. Tato aplikace slouží pro rychlou SMS korespondenci diktovanou hlasem. Obrazovka hlasového vstupu je znázorněna na obr. č. 4.9. [2]

Na webových stránkách pro Android vývojáře se nachází ukázková aplikace, která tuto službu rozpoznávání hlasu integruje. Po startu aplikace ověří, jestli je Android zařízení schopné funkci rozpoznání hlasu využívat. Aplikace rozpoznání hlasového vstupu se zavolá pomocí metody **startActivityForResult()**, které se navíc předá parametr specifikující, jaký jazykový model se má zvolit. Poté se spustí aplikace rozpoznávání hlasového vstupu a převede vstupní hlasový signál na text. Tento proces neprobíhá na samotném Android zařízení, ale požadavek je odeslán na server Google, který ho zpracuje. Rozpoznaný řetězec je pak předán zpět původní aplikaci zavoláním metody **onActivityResult()**. Co se týká jazykových modelů, jsou k dispozici **free_form** a **web_search**. **Free_form** je vhodný pro rozpoznávání delších hlasových vstupů, např. při diktování emailu nebo SMS. Pokud se očekávají kratší hlasové vstupy, je výhodnější použít model **web_search**. Takto se zajistí vyšší přesnost rozpoznání slov. Nevýhodou **free_form** modelu je, že není na rozdíl od **web_search** modelu optimalizovaný pro všechny jazyky,



Obrázek 4.9: Integrace hlasového vstupu v aplikaci Handcent SMS. [2]

které Google v současné době podporuje. Ukázkový kód integrace rozpoznání hlasu do Android aplikace je uveden ve výpisu 1. [2]

```
// Check to see if a recognition activity is present
PackageManager pm = getPackageManager();
List<Activity> activities = pm.queryIntentActivities(
    new Intent(RecognizerIntent.ACTION_RECOGNIZE_SPEECH), 0);
if (activities.size() != 0) {
    speakButton.setOnClickListener(this);
} else {
    speakButton.setEnabled(false);
    speakButton.setText("Recognizer not present");
}
```

Výpis 1: Kód pro integraci funkce hlasového vstupu do Android aplikace. [2]

4.2 SVOX TTS

Po mnoho let je firma SVOX známá jako přední poskytovatel hlasových řešení v oblasti automobilů a mobilním průmyslu, které zahrnují jak převod textu na řeč, tak rozpoznávání řeči a její převod na text. Firemní produkty našly své použití v autech, mobilních zařízeních a jiné spotřební elektronice. [15]

Z pohledu syntézy řeči na zařízení Android tato firma nabízí svůj TTS engine pro Android vývojáře, který podporuje až 25 světových jazyků včetně češtiny. Na tomto enginu je založeno mnoho Android aplikací pro čtení PDF, SMS zpráv, názvu volajícího aj. Mezi tyto aplikace, které je možné najít na Android Marketu, patří např. *Speak Text Easy* nebo *Announcify*. Pokud chce Android vývojář využít a integrovat SDK firmy SVOX do své aplikace, je potřeba o to požádat emailem. [15]

4.3 Android TTS

SDK operačního systému Android podporuje od vydání verze 1.6 (API level 6) možnost syntézy řeči v různých jazycích. Tento TTS engine je obsažen v každém Android zařízení od výše zmíněné verze a v současné době podporuje tyto jazyky: angličtina, francouzština, němčina, italština a španělština. Pro anglický jazyk zároveň podporuje dvě verze akcentů, a sice britský a americký. Dříve než je instance TTS enginu vytvořena, je potřeba zvolit, jaký jazyk se má použít. Nicméně z důvodů omezeného místa pro úschovu dat se může stát, že dané Android zařízení nemá předinstalovaný požadovaný jazyk. Pro tento případ obsahuje TTS API funkci, která automaticky vyšle platformě požadavek na stažení a doinstalování požadovaného jazyku do paměti PDA zařízení. Ověření dostupnosti požadovaného jazyka je na úrovni kódu realizováno pomocí intentu: [3]

```
Intent checkIntent = new Intent();
checkIntent.setAction(TextToSpeech.Engine.ACTION_CHECK_TTS_DATA);
startActivityForResult (checkIntent, MY_DATA_CHECK_CODE);
```

Výpis 2: Kód spuštění intentu pro ověření dostupnosti požadovaného jazyka pro TTS engine. [3]

Návratový kód odpovídající proměnné *CHECK_VOICE_DATA_PASS* signalizuje, že zvolený jazyk je nainstalovaný a TTS modul je připraven k použití. V opačném případě bude potřeba daný jazyk doinstalovat. K tomuto účelu se používá *ACTION_INSTALL_TTS_DATA* intent, který uživatele přesměruje na službu Google Play a umožní mu jazyk stáhnout. Instalace dat se spustí automaticky poté, co bude stahování jazyka dokončeno. Ukázková implementace metody *onActivityResult()* je uvedena ve výpisu 3. [3]

```
private TextToSpeech mTts;
protected void onActivityResult(
    int requestCode, int resultCode, Intent data) {
    if (requestCode == MY_DATA_CHECK_CODE) {
        if (resultCode == TextToSpeech.Engine.CHECK_VOICE_DATA_PASS) {
            // success, create the TTS instance
            mTts = new TextToSpeech(this, this);
        } else {
            // missing data, install it
            Intent installIntent = new Intent();
            installIntent.setAction(
                TextToSpeech.Engine.ACTION_INSTALL_TTS_DATA);
            startActivity ( installIntent );
        }
    }
}
```

```
    }  
}
```

Výpis 3: Kód pro stažení a instalaci chybějícího jazyka. [3]

Android SDK TTS engine obsahuje globální frontu vstupů určených pro zpracování pomocí syntézy řeči, kde každá instance třídy *TextToSpeech* může spravovat i vlastní frontu. Syntéza položek fronty se zahájí pomocí metody **speak()**, která způsobí přerušování aktuálního průběhu syntézy a umístí novou položku do čela fronty. Ukázka použití metody **speak()** je znázorněna ve výpisu 4. Nevýhodou tohoto TTS engine je, že v současné době nepodporuje Český jazyk. [3]

```
String myText1 = "Did_you_sleep_well?";  
String myText2 = "I_hope_so,_because_it's_time_to_wake_up.";  
mTts.speak(myText1, TextToSpeech.QUEUE_FLUSH, null);  
mTts.speak(myText2, TextToSpeech.QUEUE_ADD, null);
```

Výpis 4: Ukázka použití metody *speak()*. [3]

5 Návrh řešení

Cílem této diplomové práce je návrh a realizace programu pro zpřístupnění základních funkcí mobilního telefonu lidem se zrakovým postižením. Pokyny pro vypracování této práce byly stanoveny v následujících bodech:

1. Seznamte se s aplikacemi pro klienty se zrakovým postižením, problematikou analýzy a syntézy řeči, návrhem GUI pro zrakově postižené.
2. Analyzujte dostupné prostředky pro rozpoznání a syntézu řeči na platformě Android a navrhnete vhodné řešení.
3. Seznamte se s programovacím jazykem Java na mobilní platformě Android.
4. Implementujte aplikaci pro platformu Android a proveďte zhodnocení dosažených výsledků.
5. Porovnejte dosažené výsledky s existujícími aplikacemi, které se touto problematikou zabývají.
6. Vypracujte uživatelskou a programátorskou dokumentaci.

5.1 Návrh řešení analýzy a syntézy řeči na Android zařízení

5.1.1 Volba softwarových a hardwarových prostředků

Po analýze a zhodnocení prostředků pro realizaci bodů zadání této diplomové práce jsem se rozhodl použít následující softwarové a hardwarové prostředky:

- **Mobilní zařízení:** Vodafone 945 s OS Android
- **Vývojové prostředí:** Eclipse
- **Syntéza řeči:** TTS modul služby Google Překladač [7]
- **Rozpoznávání řeči:** Speech Input [2]
- **Framework:** Phonegap [10] a jQuery Mobile [9]
- **Jízdní řády:** IDOS API (online) [8]

PDA zařízení jsem si vypůjčil od katedry informatiky VŠB-TUO, abych se vyhnul jednak velké paměťové náročnosti emulátoru dodávaného společně s Android SDK a dále abych byl reálně schopen otestovat funkce odesílání SMS. Krom zmíněných důvodů jsem také chtěl, aby byla výsledná aplikace otestována na reálném Android zařízení pro ověření její funkčnosti v praxi. Jelikož společnost Google přímo podporuje a vyvíjí Eclipse modul pro vývoj aplikací pro OS Android, zvolil jsem prostředí Eclipse z důvodu oficiální podpory pro případy problémů během procesu vývoje aplikace. Jako TTS engine jsem zvolil TTS modul služby Google Překladač, protože podporuje Český jazyk a

dosahuje výborné kvality syntetizované řeči. Pokoušel jsem se také získat přístup k vývojářským nástrojům pro SVOX TTS engine, ale na mé emaily mi nebylo odpovězeno. Rozpoznávání řeči a její převod na text jsem realizoval pomocí Speech Input, protože je na Android zařízení standardně předinstalováno a pro mé účely je plně dostačující s velmi dobrou přesností rozpoznávání hlasového vstupu. Designovou a funkční část jsem implementoval s využitím frameworku Phonegap v kombinaci s javascriptovou knihovnou jQuery Mobile. Jednotlivé stránky aplikace jsou pomocí frameworku Phonegap napsány v HTML, Javascriptu a CSS. To přináší výhodu portovatelnosti i na jiné platformy než Android. Ovšem v případě mé aplikace by bylo nutné přepsat funkcionality, které jsou navíc implementovány v jazyce Java. Knihovna jQuery Mobile je použita pro design komponent a samotné přepínání stránek, které bylo možné naimplementoval v jediném html souboru. Přístup k IDOS řádům jsem zvolil online. Jednak kvůli existujícímu API a dále protože offline verze není k dispozici ke stažení.

5.1.2 Syntéza řeči

Pro syntézu řeči jsem se rozhodl použít TTS modul webové služby Google Překladač. Tato služba nabízí uživatelům primárně funkce překladu vět nebo webových stránek až ve 64 jazycích. Krom této primární služby umí Google Překladač také vygenerovat fonetickou transkripci jak zdrojového, tak cílového jazyka a taky jeho výslovnost pomocí syntézy řeči. Podpora syntézy řeči není dostupná pro všechny jazyky, ale čeština mezi ně patří.

Pro využití syntézy řeči produktu Google Překladač jsem se rozhodl z následujících důvodů:

- podpora Českého jazyka
- vysoká kvalita syntetizované řeči
- nízká výpočetní náročnost na straně PDA
- a z toho plynoucí rychlá odezva (závisí na kvalitě připojení k internetu)
- dostupné zdarma

Nevýhodou této služby je omezení vstupní věty na 100 znaků, nepřítomnost programátorského API a závislost na připojení k internetu. Z toho vyvstávají dva problémy. Prvním problémem bude vytvoření API, pomocí kterého bych byl schopen TTS modul služby Google Překladač využívat přímo z mé aplikace a druhým problémem vymyslet způsob rozdělování vět na menší části včetně optimalizace, aby nedocházelo k příliš velkým časovým prodávám mezi jednotlivými segmenty textu.

API pro TTS modul

Samotný TTS modul služby Google Překladač je dostupný na URL adrese http://translate.google.cz/translate_tts. Pro jeho využití bude potřeba

odeslat GET požadavek protokolu HTTP se správně nastavenými parametry, které jsou uvedeny v tabulce 6.

Parametr	Popis
tl	Kód cílového jazyku (<i>Target Language</i>).
q	Vstupní věta (<i>Query</i>).
ie	Kódování vstupního textu (<i>Input Encoding</i>).

Tabulka 6: Popis parametrů GET požadavku pro využití TTS modulu Google Překladače.

Krom výše zmíněných parametrů je ještě potřeba uvést do hlavičky HTTP požadavku User-Agent řetězec. Tento řetězec slouží pro identifikaci webového prohlížeče a Google jej zároveň používá pro filtrování internetových robotů. Pokud bychom řetězec User-Agent neuvedli, TTS modul Google Překladače by nám nevrátil požadovaný výsledek.

Co se týká hodnot jednotlivých parametrů, budeme jako kódování používat **UTF-8**. Pro češtinu je stanoven kód jazyka **cs**. A jako **User-Agent** řetězec zvolíme například **Mozilla/5.0 (X11; U; Linux i686; cs-CZ; rv:1.9.1.7) Gecko/20100106 Ubuntu/9.10 (karmic) Firefox/3.5.7**. Pokud bychom chtěli převést na řeč text „Test“, pak bude HTTP požadavek vypadat následovně (viz tabulka 7).

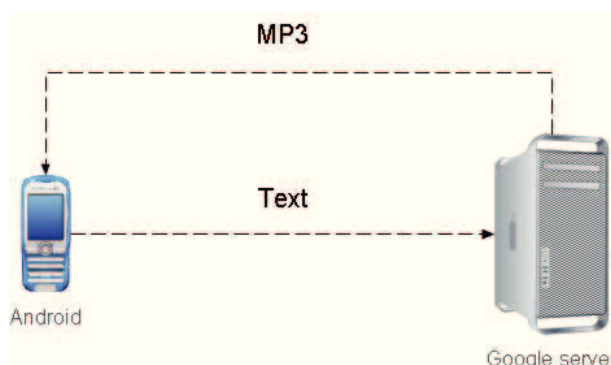
Parametr	Hodnota
tl	cs
q	Test
ie	UTF-8
Hlavička HTTP protokolu	
User-Agent	Mozilla/5.0 (X11; U; Linux i686; cs-CZ; rv:1.9.1.7) Gecko/20100106 Ubuntu/9.10 (karmic) Firefox/3.5.7
Výsledná URL	
http://translate.google.cz/translate_tts?ie=UTF-8&q=Test&tl=cs	

Tabulka 7: Hodnoty parametrů GET požadavku pro syntézu slova „Test“.

Vstupem TTS služby firmy Google je tedy kódovaný text, který chceme převést na hlasový signál. Výstupem je hlasový signál v komprimovaném souboru typu MP3. Pro přehrání tohoto souboru je potřeba ho nejdříve uložit do paměti PDA. Pro ukládání pomocných souborů jsem zvolil paměťovou SD kartu. Na jejím kořenovém adresáři aplikace vytvoří adresář *VisualAssistant* (pokud neexistuje) a s ním i další potřebné podadresáře. Pro ukládání MP3 souborů s hlasovým signálem je použit podadresář *tmp*. Schéma funkce syntézy textu v řečový signál je schématicky znázorněna na obr. č. 5.10.

Optimalizace syntézy řeči

Pro dostatečné pohodlí uživatele při používání jakékoliv aplikace je důležitým faktorem rychlost reakcí na požadované akce. V našem případě je TTS modul omezený kvalitou



Obrázek 5.10: Schéma funkce převodu textu na řeč pomocí TTS služby firmy Google.

připojení k internetu, rychlostí odezvy serveru Google a počtem textových segmentů určených k převodu na řečový signál. Technicky ovlivnitelný prvek je v tomto případě pouze kvalita připojení k internetu. Odezva serveru Google ovlivnit nelze a bude záviset především na jeho aktuálním vytížení. Počet segmentů textu k řečové syntéze je předem pevně daný, kde počet těchto segmentů závisí na velikosti celkového textu.

Co se však dá v případě časově efektivní syntézy segmentů textu udělat, je zvolit vhodný přístup ke způsobu zpracování požadavků pomocí serveru Google. Z možností se nabízí použít více vláken v aplikaci a tím nechat běžet převod segmentů textu na řeč v pozadí současně s přehráváním už převedeného řečového signálu, nebo využít techniky paralelizace a takto získat z výkonu maximum.

Sekvenční zpracování v dalším vlákne má výhodu optimálního využití připojení k internetu, kde předpokládám klasické připojení přes mobilního operátora. Tato možnost není tolik výpočetně náročná a nepřetěžuje kapacitu připojení k internetu, která je zpravidla u mobilních operátorů v porovnání s Wifi nebo ADSL omezená. Pokud bychom zvolili využít techniky paralelizace a rozhodli se požadavky převodu textu na řeč serveru Google paralelizovat v oddělených vláknech, dosáhli bychom teoreticky maximální výkon, který by však v praxi nemusel dosahovat tak dobrých výsledků jako první možnost, a to především z důvodu přetížení kapacity připojení k internetu a přílišné zátěže CPU PDA zařízení.

5.1.3 Analýza řeči

Pro rozpoznávání hlasových příkazů v prostředí frameworku Phonegap jsem zvolil Android plugin *SpeechRecognizer* [14], který budu moci volat přímo přes Javascript. Tento plugin využívá předinstalovanou funkci hlasového vstupu, která je popsána v kapitole 4.1. Rozpoznání hlasového vstupu se spustí zavoláním metody **startRecognize()**, která bere jako vstup celkem 6 parametrů. Prvními dvěma parametry jsou ukazatele na funkce pro zpracování úspěchu rozpoznání a selhání (ve zmíněném pořadí). Třetím parametrem je kód požadavku, který je předán metodě pro zpracování úspěšného rozpoznání hlasového vstupu. Podle tohoto kódu budu moci rozlišit, z jaké stránky bylo rozpoznávání

hlasového vstupu spuštěno a co se má s výsledným rozpoznaným textem udělat. Čtvrtým parametrem v pořadí je maximální počet rozpoznaných výsledků. V mém řešení jsem zvolil hodnotu tohoto parametru 1 a tedy vždy pak dále zpracovávám pouze první nejlepší shodu. Poslední dva parametry v pořadí jsou volitelné a určují text v dialogu rozpoznávání hlasového vstupu a jazyk rozpoznávané řeči.

5.2 Analýza a návrh základních funkcí programu

5.2.1 Hlasové příkazy a uživatelské rozhraní

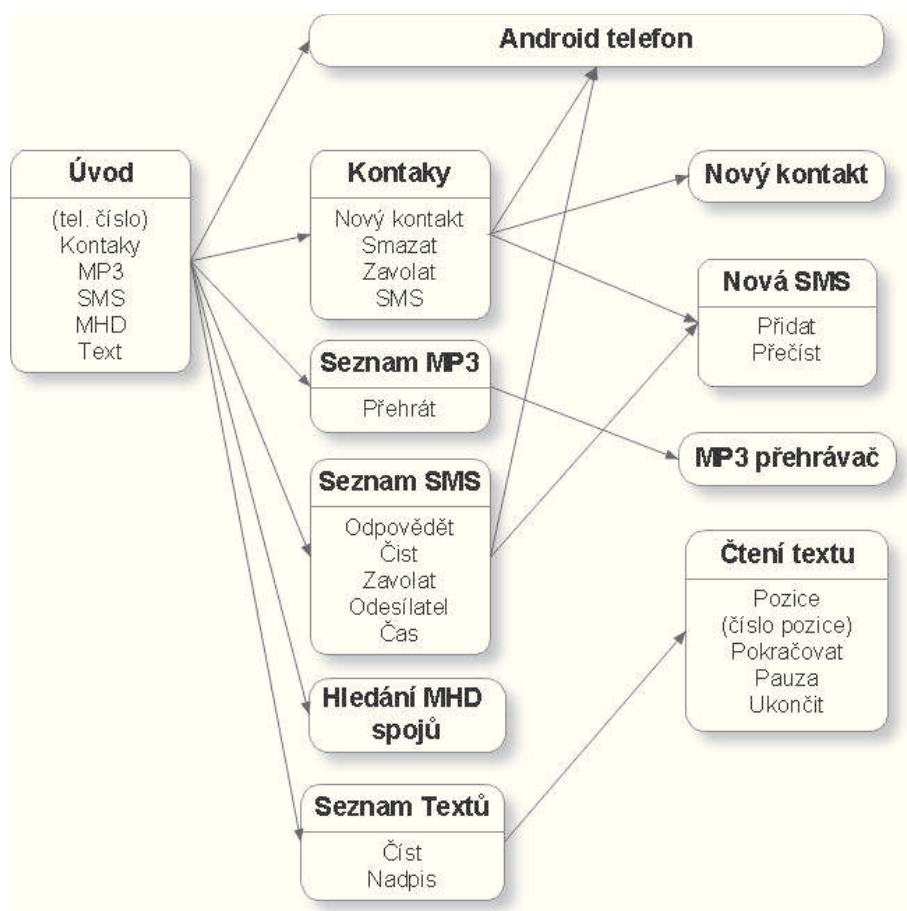
Navržená aplikace předpokládá uplatnění u uživatelů se zhoršenou schopností vidět. Uživatelské rozhraní je proto této skutečnosti uzpůsobeno volbou vhodných prvků komunikace s uživatelem a formou odezvy reakcí na akce jednotlivých prvků. Pro volbu základních operací jsem použil tlačítka, jejichž šířka pokrývá většinu prostoru šířky obrazovky PDA zařízení a text uvnitř tlačítka má maximální možnou velikost v rámci rozměrů daného prvku. Rozhraní jsem se snažil navrhnout co nejjednodušeji, a proto na každé stránce aplikace se nenachází více než 3 tlačítka. Při každé reakci na stisk tlačítka PDA zařízení zavibruje, aby se dalo uživateli najevo, že požadovaná akce započala, a dále zavibruje vždy před každou operací syntézy řeči. Vibrace v případě syntézy řeči jsem zvolil pro případ delší odezvy Google serveru, aby uživatel věděl, že má čekat.

Složitější vstupy jsou řešeny funkcí hlasových vstupů, která je spuštěna stiskem tlačítka *PŘÍKAZ*. Nejčastěji jsou hlasové vstupy používány pro přesun mezi stránkami aplikace. Struktura jednotlivých stránek a jejich příkazů je znázorněna na obr. č. 5.11.

Stránky, u kterých v obrázku 5.11 výpis příkazů chybí, hlasový příkaz nepoužívají. Krom uvedených příkazů mají stránky používající hlasový vstup navíc příkaz "Nápořád", který uživateli sdělí seznam použitelných příkazů na dané stránce, na které se aktuálně nachází. Po vykonání každé akce je vždy formou řeči oznámeno, jaká akce se vykonala. Uživatel má možnost zopakovat sdělení poslední akce a aktuální stránky, na které se nachází, stiskem tlačítka *MENU* na svém PDA zařízení. Princip čtení akcí a aktuálních stránek včetně možnosti opakování je inspirován profesionálními produkty pro zrakově postižené, které jsou probrány v kapitole 3.

5.2.2 Jízdní řády IDOS

Důležitou část života zrakově postiženého člověka představuje cestování ať už v rámci města, anebo mimo město. V mé práci jsem se soustředil na cestování pomocí prostředků městské hromadné dopravy ve městě Ostrava. Konkrétně vyhledávání zahrnuje spoje tramvají, autobusů a trolejbusů. Způsoby, jak získat informace o příjezdech MHD u konkrétních spojů jsou dva. První z nich vyžaduje připojení k internetu a jedná se o tzv. online přístup, pro který má IDOS speciální API. Druhý způsob (offline přístup) spočívá ve stažení aktuálního jízdního řádu a hledání příjezdů v něm. V mé práci jsem zvolil přístup online.



Obrázek 5.11: Schéma rozvržení a propojení jednotlivých stránek aplikace.

Online přístup - IDOS API

Pro hledání spojů mimo hlavní webový portál a pro usnadnění hledání v aplikacích různého typu má IDOS své vlastní API [8]. Pomocí tohoto API je možné snáze integrovat funkce vyhledávacího systému IDOSu do požadované aplikace. Ukázkový seznam druhů spojů je v tabulce 8.

V mé aplikaci se soustředím na vyhledávání spojů městské hromadné dopravy (tramvají, autobusu a trolejbusů) ve městě Ostrava. Pro získání dat použiji u IDOS API URL pro hledání MHD spojů Ostravy, tj. <http://jizdnirady.idnes.cz/ostrava/spojeni/>. Do samotného požadavku bude potřeba přidat ještě další argumenty, kterými určíme, jaká bude počáteční zastávka a jaká cílová. Systém IDOS pak již sám pozná, zda se jedná o zastávku tramvaje, autobusu či trolejbusu. Dále budeme chtít pouze přímé spoje, které budeme vyhledávat podle času odjezdu. Čas zadávat nebudeme, protože systém IDOS automaticky zvolí aktuální čas jako výchozí. Všechny data budeme získávat HTTP metodou GET. V tabulce

Druh dopravy	URL adresa
Jízdní řád vlaků	http://www.idos.cz/vlaky/
Jízdní řád autobusů	http://www.idos.cz/autobusy/
Kombinace Vlaků + Autobusů	http://www.idos.cz/vlakyautobusy/
Kombinace Vlaků + Autobusů + MHD Brno, Ostrava a Praha	http://www.idos.cz/vlakyautobusymhd/
Pražská integrovaná doprava	http://www.idos.cz/pid/
IDS Jihomoravského kraje	http://www.idos.cz/idsjmk/
IDS Moravskoslezského kraje ODIS	http://www.idos.cz/odis/
IDS Libereckého kraje - IDOL	http://www.idos.cz/idol/
MHD Olomouc	http://www.idos.cz/olomouc/
MHD Ostrava	http://www.idos.cz/ostlava/
MHD Pardubice	http://www.idos.cz/pardubice/
MHD Praha (bez vlaků)	http://www.idos.cz/praha/

Tabulka 8: Ukázkový výpis druhů vyhledávání spojů a jejich URL adresy.

9 je uveden seznam a popis argumentů, které budeme potřebovat pro získání odjezdů spojů MHD v Ostravě.

Argument	Popis
f	počáteční zastávka
t	cílová zastávka
byarr	hledat podle příjezdu (false)
submit	odeslání formuláře (true)

Tabulka 9: Popis argumentů pro získání výsledků hledání MHD spojů.

Například, pokud bychom chtěli vyhledat spoj ze zastávky A do zastávky B, pak by výsledná URL byla v následující podobě: *<http://jizdnirady.idnes.cz/ostlava/spojeni/?f=A&t=B&byarr=false&direct=true&submit=true>*. Uživatelské rozhraní vyhledávání MHD spojů obsahuje 3 tlačítka. První dvě tlačítka slouží k určení počáteční a cílové zastávky, a poslední tlačítko zahájí hledání spojů. Nalezené spoje jsou přetčeny formou hlasového výstupu.

5.2.3 Čas

Nezbytnou součástí každodenního lidského života je znalost aktuálního času a data. Tlačítko pro sdělení data a času jsem umístil na úvodní stránku, kde se nachází spolu s tlačítkem PŘÍKAZ. Ve výstupu hlasového signálu je nejdříve sděleno datum (den a měsíc) a poté aktuální čas (hodina a minuty). TTS modul společnosti Google automaticky nepřevádí datum do formy, kterou bych požadoval, a proto je nejdříve potřeba datum a čas správně převést na text.

5.2.4 Vytočení čísla

Aplikace umožňuje kromě vytáčení čísla uloženého kontaktu taky přímé vytočení telefonního čísla. Tato funkce je součástí hlasového příkazu na úvodní stránce. Pokud vstup nevyhovuje žádnému názvu stránky, pak je prověřen, jestli se jedná o přirozené číslo. Pokud ano, je spuštěna implicitní telefonní aplikace Androidu obsahující zvolené telefonní číslo připravené k vytočení.

5.2.5 SMS

Seznam SMS zpráv je navržen pro pohyb pomocí tlačítek *nahoru*, *dolů* a jednoho dalšího tlačítka pro hlasový *příkaz*, který nad aktuální SMS zprávou vykoná požadovanou operaci. Při přechodu na stránku seznamu SMS z úvodní stránky je ihned po přechodu přečtena část aktuální SMS, na které se nachází pomyslný kurzor. Jako dostačující jsem zvolil první čtyři slova textu pro každou SMS. Pokud uživatel přejde na další SMS pomocí tlačítka *nahoru*, anebo *dolů*, jsou stejně jako v předchozím případě přečteny první 4 slova dané SMS. Pokud se uživatel dostane na poslední položku seznamu a opět stiskne tlačítko *dolů*, kurzor se přesune zpět nahoru na první položku seznamu. Obdobně to platí pro první položku seznamu a stisk tlačítka *nahoru*. U každé SMS lze pomocí hlasového příkazu zjistit její datum, čas přijetí a odesílatele. Na SMS je taky možné odpovědět nebo odesílateli zpět zavolat. Pokud se číslo odesílatele nachází v seznamu kontaktů, je pro příkaz přečtení odesílatele použito jméno ze seznamu. V opačném případě je přečteno číslo odesílatele přímo. Další funkcí v seznamu SMS je přečtení celého textu označené SMS.

V případě odpovědi na SMS, anebo napsání nové SMS aplikace přejde na stránku pro vytvoření nové SMS pomocí hlasu. Uživatelské rozhraní má opět 3 tlačítka. Systém kompozice celé textové zprávy je založen na postupném přidávání vět pomocí rozpoznání hlasu a u každého přidaného segmentu jeho verifikace pro případ, že by v procesu rozpoznání vstupního hlasového signálu nastaly odchylky. Uživatel postupuje tak, že stiskne tlačítko *PŘIDAT* a nadiktuje větu, kterou chce k finálnímu textu SMS připojit. Poté je tato věta zpět vyslovena pomocí TTS modulu společnosti Google a uživateli je nabídnut dialog pro potvrzení přidání věty k finálnímu textu, anebo její zrušení. Dalším tlačítkem má uživatel možnost nechat si přečíst kompletní text SMS v podobě, ve které bude odeslán adresátovi. Poslední tlačítko slouží k samotnému odeslání SMS, kde pro tuto funkci jsem využil plugin *SMSPlugin* [13] pro framework PhoneGap, který je navržený pro použití na platformě Android.

5.2.6 Kontakty

Stránka seznamu kontaktů je provedená v podobném stylu jako stránka seznamu SMS. To znamená, že obsahuje tlačítka *nahoru*, *dolů* a *příkaz*. Po přechodu na stránku kontaktů z úvodní stránky je přečteno jméno a příjmení aktuálního kontaktu, na kterém je pomyslný kurzor. Stránka seznamu kontaktů dále obsahuje funkce vytvoření nového kontaktu, smazání kontaktu, vytočení čísla kontaktu a napsání SMS kontaktu.

Vytvoření nového kontaktu se spustí zadáním příkazu „nový kontakt“, který uživatele přesměruje na stránku přidání nového kontaktu. Uživatelské rozhraní této stránky obsahuje 3 tlačítka: *jméno*, *příjmení* a *telefon*. Po kliknutí na některé z tlačítek se nejdříve přečte jeho název a až poté může uživatel zadat hlasový vstup. To, co zadal, se opět znovu přečte, aby měl uživatel zpětnou vazbu ohledně úspěšnosti rozpoznání jeho hlasového vstupu. Až uživatel vyplní všechny tři údaje, je automaticky spuštěna funkce uložení kontaktu do paměti telefonu a dojde k přesunu zpět na stránku seznamu kontaktů.

Pro vytvoření čísla kontaktu se používá nativní telefonní Android aplikace, která je předinstalovaná ve všech PDA zařízeních s operačním systémem Android. Podobně jako v případě seznamu SMS se pro funkci vytvoření nové SMS využívá stejná stránka a text SMS zprávy se vytváří také stejným způsobem. Kontakt může být smazán zadáním příkazu „smazat“.

5.2.7 Čtení textů

Na stránku čtení textů se uživatel dostane z úvodní stránky zadáním příkazu „text“. Seznam textů obsahuje všechny soubory v podadresáři *VisualAssistant/txt*. Vstupní formát pro čtečku textů je kódovaný text znakovou sadou UTF-8. Záměrem této funkcionality je zpřístupnit zrakově postiženým uživatelům čtení dlouhých textů (knih, článků, apod.). Na rozdíl od způsobu čtení SMS zpráv a kontaktů v seznamu se v případě textů přečte první řádek. Z tohoto důvodu je dobré při ukládání textu do podadresáře k tomu určeného přidat první řádek s nadpisem, který daný text jednoznačně vystihuje a jeho uživatel si ho snadno zapamatuje. Uživatel si může nechat nadpis aktuálního textu zopakovat zadáním příkazu „nadpis“. V případě, že chce uživatel přečíst celý text, zadá příkaz „číst“, který ho přesune na čtečku textu.

Stránka čtečky textu obsahuje tlačítka *přečíst* a *příkaz*. Po zahájení čtení textu je možno průběh syntézy řeči pozastavit a zadat hlasový příkaz. Podstatným prvkem při čtení dlouhého textu je aktuální pozice právě přehrávaného makro-segmentu řeči. Uživatel může zadat příkaz „pozice“, aby zjistil, na které pozici se aktuálně v textu nachází. To mu může v budoucnu sloužit k přesunu přesně na tu pasáž textu, kde naposledy skončil. Druhé využití tohoto způsobu navigace uvnitř textu je jednoduché a velmi pružné přesouvání v textu směrem dopředu i zpátky o libovolný počet kroků. Přesun pozice je realizován vyslovením pouze čísla pozice. Pokud systém zjistí, že je vstupem celé číslo, automaticky provede přesun na danou pozici v textu. Příkazem „ukončit“ pak může uživatel čtení textu přerušit a vrátit se zpět na seznam textů.

5.2.8 Přehrávání mp3

Seznam mp3 souborů obsahuje všechny mp3 soubory uložené v podadresáři *VisualAssistant/mp3* a způsob fungování uživatelského rozhraní je stejný jako např. funkce seznamu SMS, kontaktů nebo textu. U mp3 souborů je přečten pouze název souboru. Tento seznam má jediný příkaz, a sice „přehrát“. Po zadání tohoto hlasového příkazu se uživatel přesune na stránku přehrávače mp3. Uživatelské rozhraní mp3 přehrávače obsahuje

tři tlačítka bez možnosti hlasového vstupu. Tlačítka mají stejnou funkci jako běžný hudební přehrávač, tj. tlačítko pro přehrání, pauzu a pokračování, a tlačítko návratu zpět na seznam mp3. Pauzu přehrávání mp3 a pokračování v přehrávání zajišťuje z důvodu úspory místa pouze jedno tlačítko, jehož funkce je přepínána podle vnitřních stavů přehrávače.

6 Zhodnocení dosažených výsledků

Výsledné programové řešení jsem úspěšně naimplementoval a jeho funkčnost ověřil na fyzickém mobilním zařízení Vodafone 945 vybavené operačním systémem Android verze 2.1 (Eclair). Realizovaná aplikace obsahuje systém propojených stránek napsaných pomocí frameworku PhoneGap a jQuery Mobile návazně s využitím technologií HTML, Javascript a CSS. Způsob komunikace s uživatelem a samotné uživatelské rozhraní používá principy obsažené v uvedených existujících aplikacích na trhu (viz kapitola 3), které jsou zaměřené na uživatele se zrakovým postižením. Kvalita syntetizované řeči je velmi dobrá a její srozumitelnost se blíží skutečné lidské řeči. Odezva serverů Google je taktéž velmi dobrá, avšak je závislá na druhu připojení mobilního zařízení k internetu. Rychlost odezvy jsem testoval na třech typech připojení k internetu: Wifi, WCDMA a GSM. Jejich srovnání je uvedeno v tabulce 10. Nevýhodou přístupu k syntéze řeči pomocí TTS modulu společnosti Google je omezení vstupu na 100 znaků, avšak zpomalení syntézy řeči vlivem spojování segmentů vět nebylo pro praktické účely zásadní.

Druh připojení	Rychlost odezvy
Wifi	velmi rychlá
WCDMA	velmi rychlá
GSM	pomalá

Tabulka 10: Srovnání připojení k internetu v závislosti na rychlosti odezvy syntézy řeči.

Funkce hlasového vstupu, která je předinstalovaná na zařízení Android, dosahuje taktéž velmi dobrých výsledků při rozpoznávání jak krátkých příkazů, tak delších vět. Velkou výhodou této funkce je podpora Českého jazyka. Vibrace telefonu a další funkce zpřístupněné pomocí PhoneGap pluginů a frameworku samotného jsou plně funkční a použitelné.

Porovnání s existujícími aplikacemi na trhu

Pokud srovnáme navržené řešení s aplikací AT&T Mobile Accessibility Lite, rozdíl bude spočívat především v rozsahu funkcí a počtu podporovaných jazyků. AT&T Mobile Accessibility Lite podporuje, krom mnoha dalších, i Český jazyk a navíc přidává funkce: nastavení budíku, možnost procházet webové stránky, zjištění pozice podle GPS, správa záznamů v kalendáři, přístup k emailu, seznamu aplikací a nastavení základních vlastností telefonu (např. vyzváněcí tón, vibrace, atd.). AT&T Mobile Accessibility Lite dále krom hlasových vstupů přidává funkci dotykové klávesnice pro vkládání textu. Chybějícími prvky aplikace AT&T Mobile Accessibility Lite v porovnání s mou aplikací je hledání MHD spojů a čtení dlouhých textů.

V případě aplikace TalkingPlaces je její hlavní zaměření vyhledávání a správa bodů zájmů (POI), kde dále nabízí taky funkci navigačních pokynů pro chůzi pěšky s využitím Google Maps. Tato aplikace neobsahuje ulehčení pro další základní funkce mobilního telefonu jako je: psaní SMS, správa kontaktů, čtení textů a přehrávání mp3. Dále tato aplikace podporuje pouze anglický jazyk a neobsahuje možnost hledat spoje MHD.

Aplikace Voice Actions slouží pro ulehčení základních úkonů na zařízení Android, ale nicméně není přímo přizpůsobená pro uživatele se zrakovým postižením. V porovnání s mým řešením obsahuje navíc tyto funkce: navigace, vytočení firmy, odeslání emailu, zobrazení mapy, otevření webového prohlížeče, vytvoření poznámky a hledání na vyhledávači Google. Příkazy jsou pevně dány v angličtině a tedy není podporován Český jazyk. Krom přizpůsobení uživatelského rozhraní a jeho interakce s uživateli se zrakovým postižením tato aplikace dále neobsahuje funkce čtení dlouhých textů a hledání spojů MHD.

7 Závěr

V této diplomové práci jsem se zabýval analýzou, návrhem a realizací aplikace určené pro ulehčení ovládání mobilního zařízení osobám se zrakovým postižením s pomocí technologií syntézy a analýzy řečového signálu. V úvodní části této práce jsem analyzoval metody používané pro analýzu a syntézu řeči pomocí počítače. V další kapitole jsem vybral některé existující aplikace určené pro uživatele se zrakovým postižením, které jsou aktuálně dostupné na trhu pro mobilní zařízení s operačním systémem Android. Způsobem realizace uživatelského rozhraní těchto aplikací jsem se inspiroval při návrhu a funkce uživatelského rozhraní výsledného programového řešení. V následující kapitole jsem analyzoval dostupné prostředky pro analýzu a syntézu řeči pro platformu Android včetně popisu jejich výhod a nevýhod. V následném návrhu jsem shrnul a vybral vhodné softwarové a hardwarové prostředky pro realizaci výsledné aplikace, kterou jsem úspěšně implementoval a odzkoušel na mobilním zařízení Vodafone 945. V předposlední kapitole jsem porovnal dosažené výsledky s existujícími aplikacemi na trhu a zmínil jejich vzájemné odlišnosti.

Výsledkem této práce je aplikace pro mobilní zařízení s operačním systémem Android, která umožňuje uživatelům se zrakovým postižením vykonávat základní funkce mobilního zařízení, mezi které patří: SMS, kontakty, čtení textu, přehrávání mp3, čas a vyhledávání MHD spojů ve městě Ostrava. Uživatelské rozhraní je přívětivé svou jednoduchostí a interakcí formou vibrací společně se syntézou a rozpoznáváním řeči. Implementované řešení dosahuje velmi dobrých výsledků co se týká úspěšnosti rozpoznávání hlasového vstupu, kvality syntetizované řeči a ostatních funkcí aplikace. Nevýhodami velké části funkcí této aplikace je nutnost připojení k internetu a omezení TTS modulu v oblasti velikosti vstupní věty, která v případě dlouhých vět může snížit plynulost syntézy řeči.

Co se týká doporučení pro vylepšení funkcí do budoucna, meze v této oblasti nejsou omezené. Mezi konkrétní návrhy na zlepšení by mohly patřit například: funkce spojené s OCR, služby geolokace pomocí GPS, zadávání vstupů pomocí Braillovy bluetooth klávesnice a rozšíření systému vyhledávání MHD spojů o další dopravní prostředky a města.

8 Reference

- [1] Android (operační systém). In: *Wikipedia: the free encyclopedia* [online]. San Francisco (CA): Wikimedia Foundation, 2001- [cit. 2012-04-29]. Dostupné z: [http://cs.wikipedia.org/wiki/Android_\(operační_systém\)](http://cs.wikipedia.org/wiki/Android_(operační_systém))
- [2] Android Developers: *Speech Input* [online]. 2012 [cit. 2012-04-29]. Dostupné z: <http://developer.android.com/resources/articles/speech-input.html>
- [3] Android Developers: *Using Text-to-Speech* [online]. 2012 [cit. 2012-04-29]. Dostupné z: <http://developer.android.com/resources/articles/tts.html>
- [4] AT&T Introduces Mobile Accessibility Lite For Blind Or Low Vision Android Users. [online]. [cit. 2012-04-29]. Dostupné z: <http://phandroid.com/2011/10/04/att-introduces-mobile-accessibility-lite-for-blind-or-low-vision-android-users/>
- [5] Code Factory: *Making mobile phones and PDAs accessible to the blind and visually impaired* [online]. 2012 [cit. 2012-04-29]. Dostupné z: <http://www.codefactory.es>
- [6] ČERNOCKÝ, Jan. *Zpracování řečových signálů - studijní opora*. [s.l.] : [s.n.], 2006. 128 s. Dostupný z WWW: http://www.fit.vutbr.cz/study/courses/ZRE/public/opora/zre_opora.pdf.
- [7] Google, Inc. *Google Překladač* [online]. 2012 [cit. 2012-04-29]. Překladač Google. Dostupný z WWW: <http://translate.google.cz/>.
- [8] CHAPS spol. s.r.o.: Možnost využití odkazu na www.idos.cz. [online]. [cit. 2012-04-29]. Dostupné z: <http://www.chaps.cz/idos-moznost-vyuziti-odkazu.asp>
- [9] JQuery Mobile [online]. 2012 [cit. 2012-04-29]. Dostupné z: <http://jquerymobile.com/>
- [10] PhoneGap [online]. 2012 [cit. 2012-04-29]. Dostupné z: <http://phonegap.com/>
- [11] PRÁŠEK, Petr. Formantová syntéza [online]. 2001 [cit. 2012-04-29]. Dostupné z: <http://noel.feld.cvut.cz/vyu/dzr/dzr12/>
- [12] PSUTKA, Josef. *Komunikace s počítačem mluvenou řečí*. Praha : Academia, 1995. 287 s. ISBN 80-200-0203-0.
- [13] SMSPlugin. [online]. [cit. 2012-04-29]. Dostupné z: <https://github.com/phonegap/phonegap-plugins/tree/master/Android/SMSPlugin>
- [14] SpeechRecognizer. [online]. [cit. 2012-04-29]. Dostupné z: <https://github.com/phonegap/phonegap-plugins/tree/master/Android/SpeechRecognizer>

- [15] SVOX Mobile Voices [online]. [cit. 2012-04-29]. Dostupné z:
<http://svoxmobilevoices.wordpress.com/>
- [16] Talking Places: *Intelligent Representation of Surroundings for Visually Impaired People* [online]. [cit. 2012-04-29]. Dostupné z:
<http://code.google.com/p/augmented-reality-android-diploma-thesis/>
- [17] Voice Actions for Android: *Control your Android phone just by speaking using Voice Actions* [online]. 2012 [cit. 2012-04-29]. Dostupné z:
<http://www.google.com/mobile/voice-actions/>

A Přílohy na CD-ROM

Příloha 1

Text této diplomové práce v elektronické podobě.

Příloha 2

Zdrojové kódy a spustitelné aplikace.

Příloha 3

Uživatelská dokumentace.

Příloha 4

Programátorská dokumentace (ve formě JavaDocu).